Project Report
On
Bluetooth Chat Application

Presented By:
Milan H. Bhatt (04)
Chirag B. Gohil (17)

Project Guide:
Keyur Zala
Pradip Vanpariya

Submitted To:
SHREE M. & N. VIRANI SCIENCE COLLEGE
Preface

Bluetooth Chat Application is useful for Chatting via Bluetooth connectivity within the limited criteria. If we have this application in our mobile, no need to send SMS. Therefore without sending SMS we can chat with another device with the help of Bluetooth connectivity which is the main use of this application. The major defect of this device is, if both of the cells are out of Bluetooth range then it’s difficult to get connection between both the devices.

Bluetooth Chat Application has two phases:

1. SERVER
   &
2. CLIENT.

Server is the machine which handles the particular client. Client is the machine which can connect to Server. By this process we can easily chat with other device. First the Server initializes and finds the other active Bluetooth device and after that process Server & Client gets connected with each other, and two devices can chat with each other.

In this way two users can chat, but there is one limitation in this application. Server is able to find all the active Bluetooth devices but it can connect with only one at a time. There is no way to chat simultaneously with all devices. Here one more drawback is limited range. We can only chat within the Bluetooth range which is limited up to certain specific distance.

In this way, Bluetooth chat application provides scintillating chatting facility. The main purpose of this application is to send text message with the help of Bluetooth tool, so no need to have sms pack in your cell!
Acknowledgement

With great pleasure, we take this opportunity to express our gratitude towards all the people who have helped and inspired us in our project work.

We are extremely thankful to all those who are extremely cooperative and helpful during this project.

We would like to express our deepest gratitude to, Keyur Zala & Pradip Vanpariya our project guide, for their invaluable guidance.

They have been a great source of help and inspiration to us in designing the project and pushing us to limits whenever we have fallen back on our project and providing new insights on the problem, we have encountered.

Milan H. Bhatt
Chirag B. Gohil
## INDEX

<table>
<thead>
<tr>
<th>Sr. NO</th>
<th>Particulars</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Project Profile</td>
<td>04</td>
</tr>
<tr>
<td>2</td>
<td>Introduction</td>
<td>05</td>
</tr>
<tr>
<td>2.1</td>
<td>Project Definition</td>
<td>05</td>
</tr>
<tr>
<td>2.2</td>
<td>What We Have Use in Our Application</td>
<td>06</td>
</tr>
<tr>
<td></td>
<td>About J2ME</td>
<td>06</td>
</tr>
<tr>
<td></td>
<td>Bluetooth</td>
<td>09</td>
</tr>
<tr>
<td></td>
<td>OBEX protocol</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Wireless Toolkit</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>Requirement Analysis</td>
<td>12</td>
</tr>
<tr>
<td>3.1</td>
<td>Hardware &amp; Software Requirement</td>
<td>12</td>
</tr>
<tr>
<td>3.2</td>
<td>Functional Requirement</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>Diagrams</td>
<td>13</td>
</tr>
<tr>
<td>4.1</td>
<td>Use Case Diagram</td>
<td>13</td>
</tr>
<tr>
<td>4.2</td>
<td>Activity Diagram</td>
<td>14</td>
</tr>
<tr>
<td>5</td>
<td>Screenshots</td>
<td>17</td>
</tr>
<tr>
<td>6</td>
<td>Bibliography</td>
<td>23</td>
</tr>
</tbody>
</table>
PROJECT PROFILE

- **Name**: Milan Bhatt
- **Roll No**: 04
- **E-mail**: milan.bhatt92@yahoo.com
- **Batch**: 6th B.C.A C4
- **Project Name**: Bluetooth Chat Application
  **Platform**: Front-end: J2ME
- **Guided By**: Mr. Keyur Zala
  Mr. Pradip Vanparsiya


Introduction

1. Project Definition:

Our project name is Bluetooth Chat Application, as the name suggests Bluetooth Chat Application is useful for Chatting via Bluetooth connectivity within the limited criteria. If we have this application in our mobile, no need to send SMS. Therefore without sending SMS we can chat with another device with the help of Bluetooth connectivity which is the main use of this application.

In this application mainly there are two phases SERVER and CLIENT. In this application there is one limitation that Server is able to find all the active Bluetooth devices but it can connect with only one at a time. There is no way to chat simultaneously with all devices.
2. **What we have use in our Project:**

**J2ME**

- It stands for Java 2 micro edition for applications which run on resource constrained devices (small scale devices) like cell phones, for example games.

- It is a stripped-down version of Java targeted at devices which have limited processing power and storage capabilities and intermittent or fairly low-bandwidth network connections. These include mobile phones, pagers, wireless devices and set-top boxes among others.

- J2ME is a reduced version of the Java API and Java Virtual Machine that is designed to operate within the limited resources available in the embedded computers and microcomputers.

- J2ME is a subset of J2SE with some API’s added specifically for the wireless devices. Just like for J2SE we have JVM (Java Virtual machine) to run the java applications; the J2ME applications run on KVM (kilo Bytes Virtual machine), which is basically a subset of JVM having limited resources, for e.g., lacking support for floating point calculations.
J2ME Technology

- J2ME is build upon the configurations, profiles and other optional packages.

- Configuration is the set of basic APIs on top of which the additional packages are built. Configuration tells you how big the KVM is and helps in the interaction with the device (via APIs).

There are two types of versions available as of now:
CLDC 1.0
CLDC 1.1
CLDC stands for Connected Limited Device Configuration.

- Profile gives you information about the mobile device and extends capabilities of configuration.
- It gives the control for the user interface, records storing etc.

There are mainly two versions of this:
MIDP 1.0
MIDP 2.0

- MIDP stands for Mobile Information Device Profile.
  One major difference between the 1.0 and 2.0 is the advanced game APIs and the better key controls support 2.0 provides.
**J2ME Architecture**

- J2ME uses configurations and profiles to customize the Java Runtime Environment (JRE).
- As a complete JRE, J2ME is comprised of a configuration, which determines the JVM used, and a profile, which defines the application by adding domain-specific classes.
- The *configuration* defines the basic run-time environment as a set of core classes and a specific JVM that run on specific types of devices.
- The *profile* defines the application. Specifically, it adds domain-specific classes to the J2ME configuration to define certain uses for devices.

- The following graphic depicts the relationship between the different virtual machines, configurations, and profiles.

- It also draws a parallel with the J2SE API and its Java virtual machine.

- While the J2SE virtual machine is generally referred to as a JVM, the J2ME virtual machines, KVM and CVM, are subsets of JVM. Both KVM and CVM can be thought of as a kind of Java virtual machine.
Bluetooth

- One of the mobile phone facilities popular right now is a Bluetooth facility, which is almost all types of mobile phones have such facilities. Bluetooth is a well-known, short-range technology for Wireless Personal Area Networks (WPAN).

- Bluetooth is a wireless communication protocol, since it's a communication protocol, we can use Bluetooth to communicate to other Bluetooth enabled devices.

- Bluetooth is like any other communication protocol that you use every day, such as HTTP, FTP, SMTP or IMTP. Bluetooth has clients-server architecture; the one that initiates the connection is the client, and the one who receives the connection is the server.

- Bluetooth is a great protocol for wireless communication because it's capable of transmitting data at nearly 1MB/s, while consuming 1/100th of the power of Wi-Fi. Bluetooth is a short-range universal wireless connectivity standard for electronic appliances and mobile devices.

- A Bluetooth connection is the result of a complex device pairing process, and provides a channel on which many data services can be provided, such as voice, internet communication, file sharing, printer connection etc.
OBEX Protocol:

- OBEX (abbreviation of Object EXchange, also termed IrOBEX) is a communications protocol that facilitates the exchange of binary objects between devices. It is maintained by the Infrared Data Association but has also been adopted by the Bluetooth Special Interest Group and the SyncML.

- The OBEX protocol provides support for object exchanges, and forms the basis for Bluetooth profile such as the synchronization profile and the File Transfer profile. The OBEX API is a separate optional package we can use either with the core Bluetooth package or independently.

- The OBEX protocol can be used over several different transmission media - Wired Infrared, Bluetooth radio and the others.

- Although OBEX was initially designed for infrared, it has now been adopted by Bluetooth. The OBEX protocol is a compact binary protocol for exchanging complex data in a simple, efficient manner.

- Bluetooth SIG [BTSIG] has defined the use of the OBEX protocol over Bluetooth connections in the IrDA Interoperability Specification. OBEX protocol, which is responsible for sending requests to a server.
Wireless toolkit

- The Sun Java Wireless Toolkit (WTK; formerly known as Java 2 Platform, Micro Edition (JavaME) Wireless Toolkit) is a state-of-the-art toolbox for developing wireless applications that are based on JavaME's Connected Limited Device Configuration (CLDC) and Mobile Information Device Profile (MIDP), and designed to run on cell phones.

- The project was initiated by Sun Microsystems team in Sun Israel Development Center in the year 2000. The developers of the first version were Daniel Blaukopf, in charge of the internals and Amir Uval on the User Interface.

- The J2ME Wireless Toolkit is a comprehensive set of tools for building MIDP applications. The toolkit can be used standalone, or incorporated into many popular integrated development environments (IDEs).

- The Sun J2ME Wireless Toolkit supports the development of Java applications that run on devices such as cellular phones, two-way pagers, and palmtops.

- Version 1.0 of the J2ME Wireless Toolkit provided a complete MIDlet build cycle at the touch of a button.

- Version 2.1 of the J2ME Wireless Toolkit implements several important enhancements, including support for JSR 185, Java Technology for the Wireless Industry (JTWI), and for JSR 172, J2ME Web Services.
Requirement Analysis

Hardware Requirement:
A Cell phone which has Bluetooth facility.

Software Requirement:
Mobile OS which supported JAVA based Application

Functional Requirement:

The main function of this application is two chat with two users very easily. This application vitally based on Server and Client. Therefore mobile can search all the active Bluetooth devices. But the limitation of this application is user can chat with only one at a time.
- **Initialization** - Any Bluetooth-enabled application, server or client, must first initialize the Bluetooth stack.

- **Client** - A client consumes remote services. It first discovers any nearby devices, then for each discovered device it searches for services of interest.

- **Server** - A server makes services available to clients. It registers them in the Service Discovery Database (SDDB), in effect advertising them. It then waits for incoming connections, accepts them as they come in, and serves the clients that make them. Finally, when the service is no longer needed the application removes it from the SDDB.
Activity Diagram:

Server
Register Service
Wait for Connections

- Initialize
- Create Service Record (SR)
- Add SR to SDDB

- Wait for Incoming Connections
- Process Incoming Connections
- Remove SR from SDDB

Client
Discover Devices and Services
Consume Service

- Initialize
- Retrieve cached devices
- Start Inquiry
- Process Discovered Device
- Process Inquiry Completed

- [If there are new devices]
  - Discover Services
  - Process Discovered Services

- [Service Found]
  - Consume Service

- [Service not Found]
  - Do nothing.

- [There are no new devices]
  - Do nothing.
You see in this figure that both client and server perform initialization, that the server application prepares a service and waits for connections, and that the client discovers devices and services, then connects to specific device to consume a particular service.

Now let's drill down into some sample code that shows how you implement these activities using the Java APIs for Bluetooth.
Discovering Nearby Devices and Services
Using Screen shot, viewers can get pictorial idea about the application.

So, here take a few Screen shot of this application.

Above figure display, when application run first time.

After pressing Launch button application will turn on.
This is the first form of our application, here ask the user if he/she want to connect with other device or not.

If user press YES button user connect with other device.
If user doesn’t want to connect with someone then user can press NO button.
Here, user can get the Menu. Menu has Connect button & Refresh button.

Using Refresh button, user can search the all active Bluetooth devices.

Connect button for the connecting to other device.
You can see in above figure first device is actually detecting second device
After detecting second device first device confirm the acceptance and then finally both the device connect with one another.
Above screenshot display the actual chatting form.
After pressing send button one can actually send the message to another device.
Textbox available for typing message.
This is the final screenshot; one can observe from the above picture that one device is actually sending data or message to another device.

Screenshot displays that the second device is getting message from the first device and the conversation is going on with the help Bluetooth connection.
Bibliography:

1) Complete Reference
2) www.wikipedia.com
3) www.roseindia.com